

# Milestone 1 Report

---

## 一、任务综述

### 1.1 小组名称

xxyyz

由各位组员各出一个字母组成。

### 1.2 课程项目综述

- 项目目的

优化19级软件需求工程课程的“代码敲得都”队所编写的大作业，完善并扩展其功能，美化其界面。

- 项目名称

开源项目分析平台

- 项目简介

旨在帮助开发者通过单一平台对不同托管平台上的开源项目进行项目分析、比较，以更好的进行项目管理。

## 二、任务描述与角色分配

### 2.1 任务描述

#### Management

任务如下：负责主导项目计划和进度安排，统一组织项目中的各个环节，对项目进行总体监管；审查和批准项目中的重要技术决策；协调组织小组人员进行各项工作，保证小组成员间交流通畅。

#### Development

任务如下：制定项目计划，探讨项目可行性，预估完成项目的开发进度；软件设计。构造设计模型，进行质量评估和改进；编码。

#### Testing

任务如下：制定测试策略；进行测试工作，撰写测试报告；与Dev讨论问题解决方案。

User Representatives

任务如下：深度调研用户需求，从用户角度评估产品，做出相应评价和提出改进意见

Paper work

任务如下：负责报告和文档的书写

2.2 角色分配

我们组一共有五位成员，可以与上述任务职责一一匹配，经组内协商后将角色分配如下。

成员姓名	任务
黄亦霄	Management
陈奕宇	Development
江小小	Testing
陈思全	User Representatives
姚逸飞	Paper Work

三、项目管理

3.1 组织架构

项目采用分布式管理模式，组长为陈奕宇，所有五名组员均担任且只担任一项工作的负责人，组织、实施、验收所负责部分的工作，具体分工及职责参见成员组成章节。

3.2 会议与日常交流

- 【平台】项目采用微信群聊作为会议和日常交流平台。
- 【会议安排】会议分为常规会议和特殊会议两类。常规会议一般由全员参与，时长为30分钟、频率定为每周一次，在没有调休安排的周六上午由管理负责人或其指派的其他模块负责人，提前至少一天进行预告和主持。特殊会议与会成员不定、时间不定，可以作为常规会议未尽事项的具体研讨，或作为项目里程碑、大版本迭代等重要事件前的最终审核，由会议主持者自行组织。

【会议管理】会议应当简洁高效，由主持人提前拟定议程、发布会议准备相关任务、发出通知并在会议开始之前将议程交给。两类会议一般均由文档负责人作记录，并在会后会议纪要，并上传公共仓库。

【日常交流】日常交流中应当保证效率。文档写作采用markdown格式，在群聊中需要他人评审的，需导出为.pdf格式并合理命名，包含阶段、版本、任务。在技术交流中涉及代码的，原则上使用新分支的方式上传至小组仓库。确实较短的，可以采用含有文件和行数的屏幕截图及源代码文本。

### 3.3 进度负责机制

【任务发布与接受】组件负责人发布任务时，需要明确截止时间、完成要求、提交方式等，任务接受者应当在包括但不限于公开或私下询问、查阅技术文档等方式充分理解之后再行接受，一经接受即须对任务负责。

【任务的异常处理】当遭遇困难，任务接受者首先应当尝试自行解决；解决不成的向组件负责人反映，有必要的可以召开特殊会议。在截止时间前的合适时机，任务发布者可以发布提醒。原则上截止时间不作延长，有确实困难的可以挂起个别任务行部分验收。

【任务的验收】任务验收由任务发布者组织，验收不通过的应当交由任务完成人修正，或进入异常处理程序召开特殊会议协商。

【进度的把握】管理负责人应在日常会议上报告预期进度及完成情况，必要可以分模块由其负人介绍。

### 3.4 工程版本控制

【平台】项目采用GitHub平台进行版本控制。由陈奕宇创建并管理私有仓库，并邀请其余全部成员共同开发。

【分支】master分支由开发负责人掌握最高的管理权限，负责审核其余分支的合并请求。开发阶段应在本地完成，并通过每个成员的独立dev-`<name>`分支进行存档。当模块通过本地验收，应当发起合并请求，由开发负责人审核通过后准予合并入master分支。

【提交】开发阶段有阶段性成果，即需要进行向个人dev分支的提交。提交信息需要写明本次提交所属的类别，如：

**feature**：新增功能；**fix**：修复bug；**docs**：修改文档；**refactor**：代码重构，未新增任何功能和修复任何bug；**build**：改变构建流程，新增依赖库、工具等（例如webpack修改）；**chore**：非src和test的修改引入一些第三方库等**test**：测试用例的修改；**ci**：自动化流程配置修改；**revert**：回滚到上一个版本；**feat**：完成一些功能；**style**：样式修改；**perf**：一些优化重构；

【版本】采用三段式，v<版本>.<里程碑>.<序号>，如v1.3.1

- 架构升级或架构重大调整，修改第2位
- 新功能上线或者模块大的调整，修改第2位
- bug修复上线，修改第3位

## 四、项目选择报告

## 4.1 pros

- 配置较为简单，基本功能完备，可以进行项目搜索并展示主要信息，以折线图展示近3个月的commit信息，以饼状图展示贡献者commit占比，以柱状图展示issue创建和解决数量，为贡献者提供访问接口,基本满足源码分析的基本要求。
- 前端技术（均为组内成员较为熟悉的技术）
  - html
  - css
  - javascript
- 后端技术（均为组内成员较为熟悉的技术）
  - mysql
  - Python Django

## 4.2 cons

- 从github提供的api爬取数据，与老师上课要求存在些许违背。
- 提示信息不完善，如在搜索框输入为空、url错误时，将直接跳出浏览器报错，观感较差。
- 或因项目完成后测试不充分，输入不同仓库url后时有bug。
- 菜单编排不合理，右上方菜单的contribution为静态页面，而事实上理应在输入url之后才能显示contribution，该按钮不应出现在菜单中。
- 图表坐标轴设置奇怪，列出了较多user，但折线图上只呈现了远少于列出数目的user，且没有对应关系。
- 页面较为简陋，不见美化痕迹。

## 4.3 预期效果

- 修复代码中可能存在的bug，使其对所有仓库都能正确运行并分析，对上述问题进行合理修改。
- 美化界面，可参考其他项目的ui。
- 增加pull request信息分析。
- 扩展代码分析数据源gitee，已预留接口。
- 图表支持多种交互、项目对比、已添加项目的储存与管理（有待根据后期项目进度规划实施部分）。

# 五、项目预期时间表

## 5.1 milestone

1. Team Workflow & v1.0 Selection（已完成）

## 2. Vision and Scope & Feature Roadmap

Vision and Scope 选 project2 时已大致确定，Feature Roadmap 需在 10.13 之前完成。

## 3. Software Requirements Specification & Mid-Term report

考虑到对代码结构和需求进行分析的时间，需在 11 月之前完成 Software Requirements Specification，用 3~4 天完成 Mid-Term report。

## 4. Design and Coding

设计和写代码作为project的主要部分，将会占据大部分时间，预计使用11.5开始的25天完成项目代码的优化和扩展，遗留十天用于项目的测试和验证。

## 5. Final Report & Final Presentation

从12.10开始，需要7天左右合作撰写最终报告，再用3天左右根据报告制作展示的ppt。

# 5.2 time table

Time table	
①Team Workflow & v1.0 Selection	10.3~10.7
②Vision and Scope & Feature Roadmap	10.8~10.13
③ Software Requirements Specification & Mid-Term report	10.14~11.3
④Design and Coding	11.4~12.8
⑤Final Report & Final Presentation	12.10~12.20